

AFM Tip Impulse Response in Attractive and Repulsive Regimes



Garrett Fitzgerald

Union College Computer Engineering '18

ECE-499 Senior Capstone Final Report

Co-advisors: Professors Catravas & Dovidenko

March 2018

Report Summary ***

This report is a benchmark of progress in the design and implementation of my senior project. The study to be completed by the end of winter term 2018 is the creation and validation of a software module for Union's atomic force microscope. This module will make possible a systematic study of the system impulse response. The design aspect of this project is almost entirely rooted in working with and customizing software which controls the AFM. At this moment in time at the end of Fall term 2017, the module is nearing completion. With the design of the project formalized, this design report can be accurate in describing the holistic design approach of the past two terms. The results of efforts in ECE 497 and 498 are culminated in a custom software module, the design and completion of which will allow for data collection and analysis of AFM tip impulse response.

This paper outlines background information about atomic force microscopy, focusing on the components which are to be accessed and manipulated by the designed software module. Motivations for creating this software are explained along with expected results. The methodology and completed strategy of designing functions which complement the existing abilities of the AFM are discussed. The creation of this software and the methods in which it will be employed to study the impulse response of the AFM system are explained. The most essential and daunting design specifications were satisfied this term, with the successful creation of a suite of programs which output various waveforms to the AFM tip and record the response of the system. Future goals and a plan for implementation with the new functional possibilities of the system are organized and presented.

Table of Contents ***

0	Introduction	4
1	Background	
5		
	1.1 AFM Overview.	5
	1.1.1 Microscope Stage and Head	6
	1.1.2 Control Electronics	
8		
	1.1.3 Computer Software	8
	1.1.4 Attractive and Repulsive Regimes	9
	1.2 Impulse Response Study- Goals and Motivations	10
	1.3 Potential Impacts of this Project	
.11		
	1.3.1 Sustainability	11
	1.3.2 Social and Microscopy Community Impact	
12		
	1.3.3 Economic Impact	12
	1.3.4 Ethical Questions	12
2	Design Requirements	14
	2.1 Asylum Software	14
	2.1.1 Low Level Software: MFP-3D Controller	15
	2.1.2 High Level Software: MFP-3D and IGOR	15
	2.2 Software Module	16
	2.2.1 Parameter Requirements	17
	2.2.2 Reproducibility and User Guide	18
	2.3 Functional Decomposition	18
	2.3.1 Code Organization	18

	2.3.2 Selection of Design Criterion	19
3	Design Alternatives	20
	3.1 Tip Response Input Data	20
	3.1.1 LVDT Voltage	
.21		
	3.1.2 Optical Laser Photodetector	21
	3.2 Excitation Strategies	22
	3.2.1 Sonification	22
	3.2.2 Choice of Excitation Output	23
4	Preliminary Proposed Design	
24		
	4.1 Software Design Breakdown	24
	4.2 Testing and Characterization of System	25
	4.3 Project Timeline	25
	4.4 Parts List	26
	4.5 Status of Module Design	27
	4.6 Completed Code and Examples	29
5	Conclusion	
32		

Section 0: Introduction***

Atomic force microscopy offers valuable insight into nanoscale material science investigation. Union College owns an Asylum Research MFP-3D atomic force microscope (AFM), located in the IRIS lab (Interdisciplinary Research Instructional Suite). Union's AFM is utilized by various disciplines for material science research, but has not yet been interrogated for its ability for operation under user defined software functions at our school. The proprietors of the system, Asylum Research, provides a rich code base and documentation behind their company's software which is open source in its ability to be used and modified by users. This offers the potential to write custom software governing control of the AFM through additional means other than the provided programs which allow for normal system operation such as imaging material surface properties. This project's objective is to validate potential to create user defined AFM software for use in the study AFM probe impulse response.

The Asylum codebase, while thorough in its provided capabilities, does not possess routines which can be used to measure the impulse response of the system's tip. Various excitation impulses output to the cantilever and tip of the microscope's probe will allow for the recording of the system's response to the impulse. However, there are no provided means to produce and record these signals. The capability to write user defined functions is a boon to the Asylum AFM system, and the design of a software module to study the impulse response is the main focus of this project. The impulse response will be studied as a function of height above the sample with linear incremented steps in amplitude of the impulse. With different z direction displacement from the sample at an atomic scale, the AFM tip experiences an attractive force to the sample at distances where Van der Waals forces take effect, and at closer distances the tip experiences repulsive forces due to the Pauli exclusion principle. It is suspected that the system response to increments in input amplitude of the signal will result with a nonlinear impulse response. The goal of this project is to design and implement software which will allow for the systematic study of this system response in attractive and repulsive regimes. With this software

designed, there will exist the ability to study a quality of atomic force microscopy which has not yet been interrogated in any substantive literature.

This report will first outline the basic background of atomic force microscopy and the interest in examining the impulse response in attractive and repulsive regimes. Different issues such as sustainability, social impact, and economic impact which this project may face will be discussed. Next in the design requirements section, an overview of the system's low level and high level code will be explained, particularly in regards to how writing custom software to control the cantilever and tip is implemented. The software suite which has been in development this term will be decomposed, with the capability of the new module functionally explained. Finally, a holistic and detailed proposed design will be presented. Included in this final section is organizing diagrams and code from the module created, preliminary data collected, and citations for how the Asylum codebase was utilized in the software's creation. This design proposal serves as the body of work which was completed this term, with the design of my ECE-498 project requiring the design to focus on creating a new software module. It is the stepping stone for beginning the implementation in the winter term of the newly created module and the study of AFM impulse response.

Section 1: Background

1.1: AFM Overview

Nanoscale research possibilities made possible with atomic force microscopy are extensive. Using the Union MFP3D Atomic Force Microscope, this project aims to explore potential applications of applying an impulse to the AFM tip. An AFM (atomic force microscope) can achieve resolution on an atomic scale down to fractions of a nanometer. For such a powerful device, its instrumentation is relatively simple. This imaging is made possible by using a probe to scan the material surface in a raster manner in x and y dimensions using piezoelectronics. The cantilever and tip of the microscope map the surface topography in the z

dimension in different scanning modes, with surface-tip distance control extending down to a single angstrom. See Figure 1 below (Eaton, West, 2010) for a comparison of different scale lengths achievable with various microscopes. Atomic force microscopy offers resolution of material surface at a higher resolution than other nanoscale imaging systems.

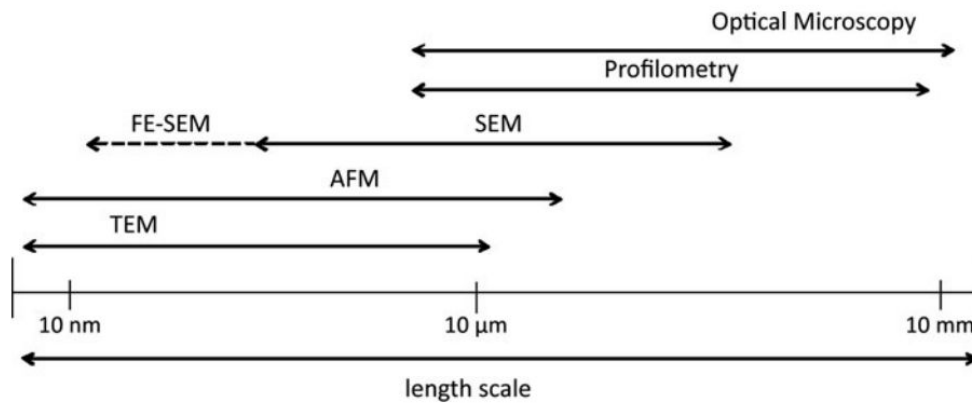


Figure 1: AFM Resolution compared to various microscopes

The basic operation of the AFM can be parsed into three chief sections of instrumentation: the microscope stage, the control electronics, and the computer. The stage contains the scanner, sample holder, and force sensor. The control electronics govern the scanning of the sample in x, y, and z dimensions with fine control piezoelectronics. These control electronics also record the position of each dimension. The computer and associated software runs the system in different modes of scanning and records the output of the sensors, compiling an image. [1]

This project is not geared towards using the AFM for its usual nanoscale imaging purposes. The property of cantilever and tip impulse response is to be studied as a function of height above the sample. This study is not currently possible with the current software provided from Asylum Research. The open source capability provide the means with which to create custom software to govern fine control of AFM components. In this vein, the basic operation of the AFM will be presented as it pertains to the overarching subject at hand. The more important background information pertains to the AFM software and control electronics. These will be

briefly introduced and later explained in more thorough detail in specific regard to the project after the elementary facts about atomic force microscopy are presented.

1.1.1: Microscope Stage and Head

The physical components which comprise the AFM are the microscope stage and head. The stage holds the sample and controls x and y position of the probe above the sample. The head holds the probe, made up of the cantilever and tip. In order to measure z dimension position, a superluminescent diode (wavelength ~ 860 nm) is positioned to reflect light against the cantilever to a position sensitive diode (photosensor). Based on the intensity of the reflected laser, the photodiode has a corresponding voltage value which is used in a feedback loop to control the set height of the z dimension piezo. Note that all scan axis are movement controlled by individual piezos and have a voltage value which correspond to position, measured by a proprietary Asylum linear variable differential transformer (LVDT). It is important to consider that there are thus two options to record z-dimension movement of the AFM cantilever and tip: the photodiode recording and the z piezo LVDT. See Figure 2 below for a diagram detailing the component interactions which record and set cantilever and tip position relative to the sample. [2]

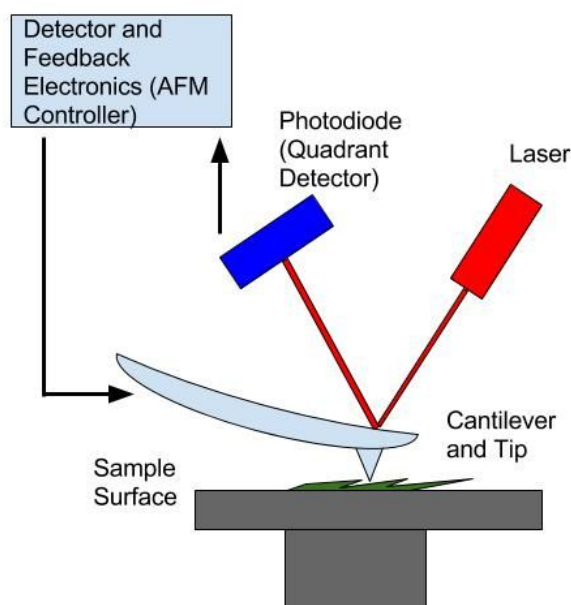


Figure 2: Basic diagram of AFM operation.

Figure 3 below shows an image of Union's AFM with the important parts labeled. As a function of x and z position, the sensors determine the movement of the cantilever and tip, providing the surface profile of the sample in the z dimension per x,y position. Through means of the photodiode and z piezo LVDT, the movement of the tip and cantilever relative to z dimension height above the sample can be detected. Additionally, an optical microscope allows for the accurate placement of the superluminescent diode on the tip of the probe.

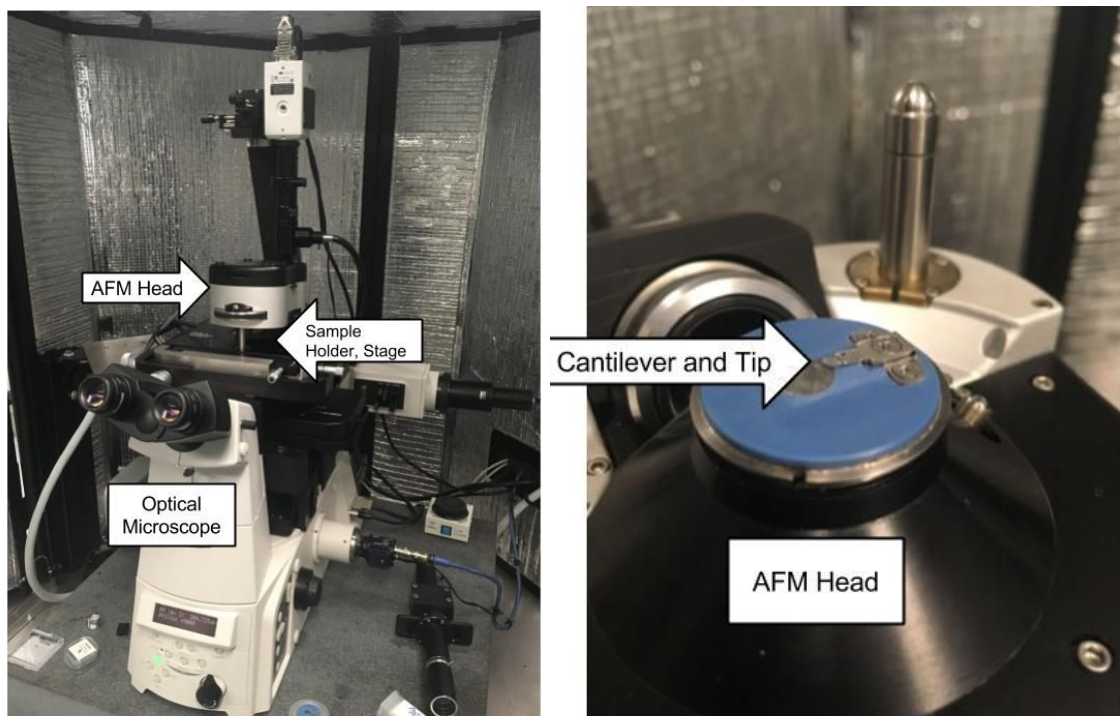


Figure 3: Union College's Asylum MFP-3D AFM. The left image shows the entire system. The right image shows the AFM Head, which contains the components to be studied.

1.1.2: Control Electronics

The Asylum AFM Controller contains a DSP and FPGA which control AFM head and stage components with analog signals. Voltage values output from the controller move scan axis and probe position. The feedback loop maintains the probe's position relative to the surface material. The control electronics are a mediator between computer software defining the

scanning function and the components themselves. Digital voltage data is given from the computer software and converted to an analog signal by the controller, and analog sensor readings from the head and stage are converted to digital data for the computer software. The A/D and D/A converters operate at 100 kHz. The controller interfaces with the control computer via a usb connection. [3]

1.1.3: Computer Software

The software driving the system is proprietary to Asylum Research. It defines the mode of scanning and imaging and collects the data from the controller. Currently running Windows XP, the computer which runs the Asylum software is not connected to the internet in order to avoid unwanted updates. The main program runs on the data analysis software environment IGOR Pro. The user interfaces with control of the AFM through this program, using a graphical user interface to choose different scanning modes and record data. The high-level IGOR code which is under the hood is not a popular programming language, however the documentation behind its capabilities are provided online by WaveMetrics, the service provider. This high level IGOR code handles data manipulation and overall control of the system, interfacing with low-level AFM controller commands. The relationship between high-level and low-level controller programming capabilities will be discussed in much more detail in the Design Requirements section.

1.1.4: Attractive and Repulsive Regimes

Depending on displacement of the tip to the surface of the material, intermolecular forces and atomic quantum forces act upon the tip. A main focus of this project is to study the impulse response of the AFM tip in two specific regions which are dependent on z-dimension relative surface distance. As the cantilever and tip approach the surface, it begins to experience attractive forces towards the surface of the sample. The attractive regime begins around ~ 10 angstroms from the tip of the surface and is due to Van der Waals forces. This attractive force is a result of an instant polarization between the interacting atoms. As the tip approaches closer to the material surface, it begins to experience deflection from the material, also known as repulsive forces. The

repulsive regime is ~ 0.1 angstroms from the surface and is due to the Pauli exclusion principle. Simply put, this repulsion from the material is due to the relative vicinity of the interacting atoms and their overlapping electron orbits at atomic distances. See Figure 4 below for a force curve that results from the extension and retraction of the tip to the surface of the material. In attractive mode, these atomic forces bend the cantilever towards the surface material, and in repulsive regimes the cantilever is bent away from the surface. [4]

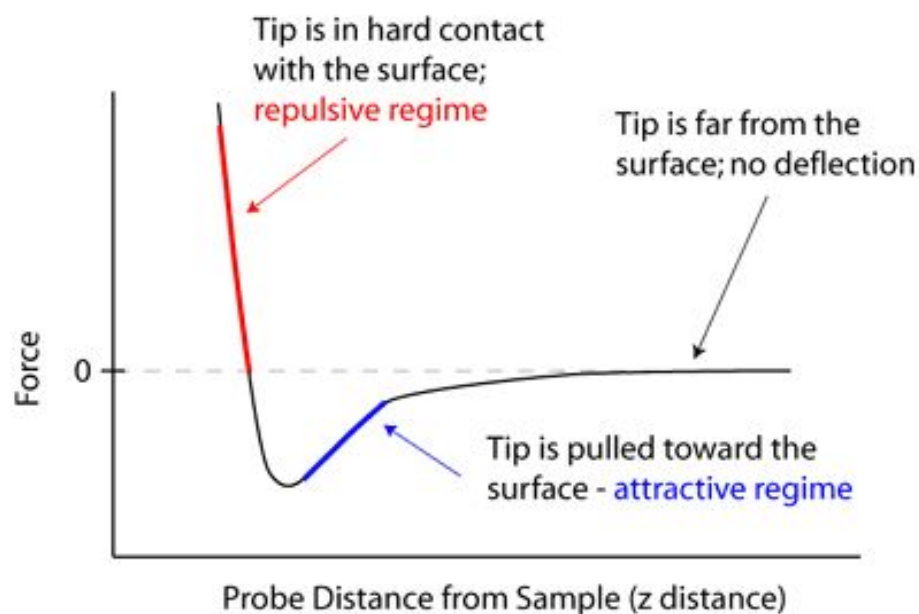


Figure 4: Force curve from Eaton, Peter, and Paul West. "Atomic Force Microscopy." (2010)

1.2: Impulse Response Study- Goals and Motivations

Prior work on the Union AFM has shown that the system response in attractive and repulsive regimes is nonlinear. Matthew Manning '12 completed a senior project, titled "Coupling of Audio Signals into AFM Images" investigated the impulse response to an input signal of audio excitations. Impulses of single frequencies and other musical signals allowed the characterization of the system's response in attractive and repulsive regimes, with results showing nonlinearity in the force-distance curve. This project used external means of applying an impulse to the AFM tip through sonification, my project aims to study the same response of

the system using user defined output waveforms. In both cases, it is in the interest of an AFM user to know more about the system which they are using, namely the effect of small signal response models when interpreting image results. The work of Matt Manning as well as my continuation in this pursuit provides the potential for AFM users to systematically characterize this information. The suspected nonlinear system measurement will reveal more information about sample environment and the overall imaging process. [5]

There is little work outside of a past Union senior project which explores the impulse response of AFM probes. According to the work of Rabe, Turner 1997: “Acoustical vibrations of atomic force microscope cantilevers can be excited either by insonification of the sample or by vibration of the clamped cantilever end. The resulting dynamical system is complex and highly nonlinear.” This paper, titled, “Analysis of the high-frequency response of atomic force microscope cantilevers” is one of the only pieces of literature which consider the impulse response of an AFM probe. A linear model for cantilever response is outlined and shown analytically to be incorrect in the modeling of the system response. The author attributes a possible reason for this nonlinearity in the system to the nonlinear nature of the atomic forces which cause the attractive and repulsive force on the tip. To investigate this suspected complex and potentially interesting issue, my design will produce a system with which the impulse response can be recorded. It is likely that with the module implemented, a thorough study of the system will be made possible for the winter term. [6]

1.3: Potential Impacts of this Project

Aside from taking steps towards exploring a relatively unknown system response of the AFM, the impacts of this project provide concrete affirmation that the Union AFM can be used for custom experiments involving user defined functions. Adding the ability to customize functionality of the Union AFM can add currently unavailable ability to future studies involving the instrument. Continuation of the study once the module is put in place is a likely route, as most of my time is spent in the writing of the code. Different engineering questions such as sustainability, social impact, economic impact, and the associated ethical questions are to be discussed.

1.3.1: Sustainability

This project is sustainable as long as Asylum Research maintains their open source code policy. Their current programming environment and community (the Asylum user forums) provided by the makers of the MFP-3D AFM allows for a module such as the one to be implemented to exist in parallel with normal AFM operation without jeopardizing any of its provided functionality. The system includes a user demo which introduces the concept of programming within the AFM software's framework along with extensive documentation for how to interface high level user interface code with the controller's machine code.

In addition to completing this module, I am creating a programming guide to be used by future Union students in their pursuit to complete a project similar to my own. Diligent documentation as well as insights from trial and error make this document invaluable for future design of user defined functions on the AFM. The code will not need maintenance as long as updates from Asylum do not change subroutine functionality.

1.3.2: Social and Microscopy Community Impact

The social and AFM community implications of this project are optimistic and open. If the results of the impulse response prove to be valuable information for AFM users, then my module may be eventually available for use along with other Asylum software (at least within the collaboration circle of this project). The user guide which teaches how to write and compile user functions adds to a small pool of community knowledge on the subject. Future research at Union may be able to benefit directly from this project at the very least from its showcase of the Asylum MFP-3D's customizability for future studies.

1.3.3: Economic Impact

The economic impact of this project are not immediate. A large problem that is inherent with AFM imaging is the breaking of probes and tips by allowing the probe to interact too harshly with the surface. With the success of the module and data collected about tip impulse response as a function of height, it will be easier to understand material surface properties when

first beginning the study of a given material. This information provided from my project can allow an experienced AFM user to reduce the number of tips which break over the course of interrogating material surface properties. This has the potential to save a large amount of money in materials science research with the AFM, as each tip costs \$10-\$80 depending on material circumstantial requirements.

1.3.4: Ethical Questions

There is an ethical grey area surrounding my project in regards to the proprietary rights of AFM software. It is the hope of myself and my advisors that the module designed for this project will open the doors to a large breast of potential new research. If the impulse response as a function of height can be quickly characterized by means of a simple function call, new information about a wide variety of sample materials may be available. If this information is valuable to the user for reasons such as avoiding breaking tips and revealing new nanoscale properties of the material, then it would be in the interest of myself and collaborators to continue this research with motivations to publish new and interesting findings. The software is open sourced, and in the case of Asylum's policy on writing user defined functions, it is entirely up to the user to use their product to do what they wish with these custom functions. After all, Asylum Research is a company which must make a profit to stay in business, so for them to provide the unique ability for users to add to their software as the user sees fit may just be a way to generate business. However, if a module were produced which proved economically/ commercially viable in their codebase, the question of ownership of this module would cause an ethical dilemma.

For this reason, I have not posted specific questions to the Asylum User Forum which could have helped me along the way. All of the code which I have written has been on my own intellectual accord, and the idea behind the pursuit of this research originates with my advisor. This would validate using Asylum's codebase as the jumping off point into deeper studies of their system. Conversely, the true spirit of open sourced software is rooted in the concept of 'free' software. That is, if a software module can add beneficial value to the scientific community, it should be shared. The resolution to this ethical question is to continue with the idea of ownership of my own work, with the actual code not being available. An alternative

would be for the logic of the program be available for those who wish to complete the same study, but this would have to be considered in more detail after data collection begins.

Section 2: Design Requirements

This section will detail the requirements of various pieces of software which are required for the study of AFM tip impulse response. The largest limitation of this project is the confinement of custom software to work with a limited amount of AFM controller low level functionality. In other words, there is a relative scarcity in freedom of programming in any language or system other than what Asylum Research already has in their codebase. The program suite that I create must work in the language of what already is established by the company, with added functionality limited to the style and standards of Asylum Research

programming organization. The signal delivery diagram which the software module must monitor is shown below.

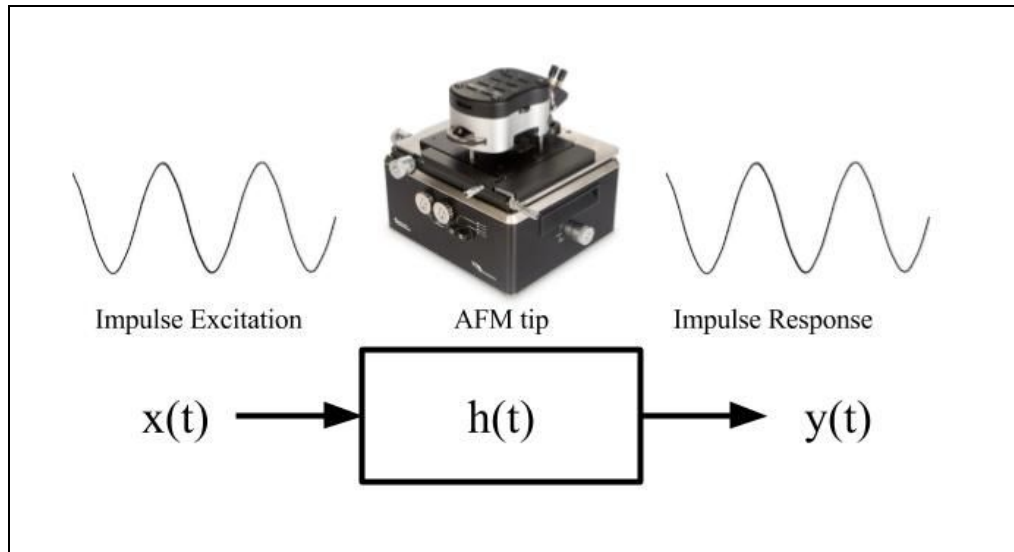


Figure 5: Signal delivery and measurement

2.1: Asylum Software

The computer running AFM software has a main program which opens the user interface for controlling imaging materials. From this main program, normal operation of the machine is relatively straightforward. There exists an extensive user guide, and most imaging techniques are well defined and easy to access. I must be able to have my module written in this convention. The calling of my impulse response measuring module should not alter normal operation of the AFM. An output waveform defined by the user should be able to be used as an impulse on the z piezo. The response of this excitation should be measured by the LVDT sensors and the photodetector quadrant detector. This process involves working with the low level AFM controller code as well as the high level IGOR code which interfaces the computer and the controller.

2.1.1: Low Level Software: MFP-3D Controller

In order to control components of the AFM head and stage, the AFM controller needs to read in a digital signal from the software, convert it to analog, and send it out to the specified components. The sensor recording of this output waveform is read in as an analog signal and digitized, in turn sent back to the computer. The term ‘low-level’ code refers to the built in functions which Asylum provides for accessing AFM controller I/O. All of these machine-interfacing routines cannot be added onto, with their parameters and control flow being the only manipulative qualities to create custom programs to control the AFM. Asylum low-level functions should be able to provide all necessary requirements for outputting a waveform to the cantilever as the impulse. The user should be able to define the waveform in the IGOR software. A low-level function should be able to record the cantilever response to this output waveform, reading it in and converting it to discrete values for the computer high level software. A diagram of the low level functions interacting with AFM components is shown below in Figure 6.

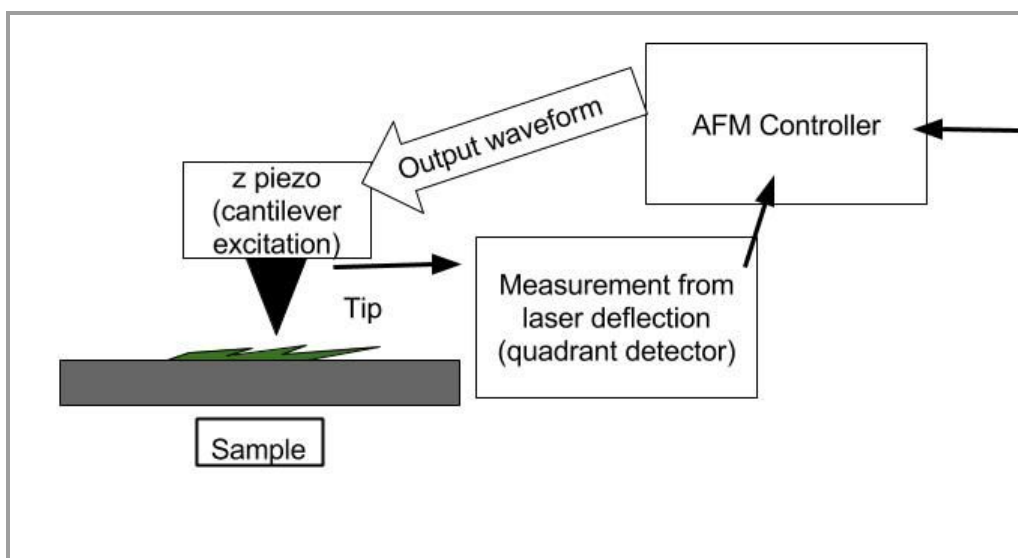


Figure 6: Low-level controller and component interaction

2.1.2: High Level Software: MFP-3D and IGOR

Within the Asylum MFP-3D IGOR Pro interface, a set of user defined functions must be able to be written, compiled, and run with the AFM in normal operating conditions. The high level code should be able to handle this delegating task of defining output waveforms and calling

low level code to send the impulse to the AFM. It should also record the input data from the controller. There should be the ability to define a chirp, kick, noise, and square waveforms in separate subroutines. Each different type of impulse will be able to be output and recorded while the AFM is in normal operation so the system response can be recorded as a function of height in attractive and repulsive regimes. Capabilities such as MATLAB data import and tabular data output for use in analysis are essential for the IGOR code to handle. Graphs should be produced of the output impulse waveform and input system response. To accommodate for large amounts of data collection, all of the input data from the response should be saved in an organized manner. Error messages should be included in every routine.

2.2: Software Module

After describing the expected capabilities of the low level and high level code, the parameters that specify output and input data with respect to the module's design will be outlined. The software module that is to be designed must be well documented and hierarchical in design to allow for easy implementation of various output waveforms and small changes in data manipulation. The following figure shows a high level outline of the software module with respect to its requirements- to output an excitation and input the system response. The high level IGOR code and the low level built in functionality must interact seamlessly in order to drive and measure the impulse response.

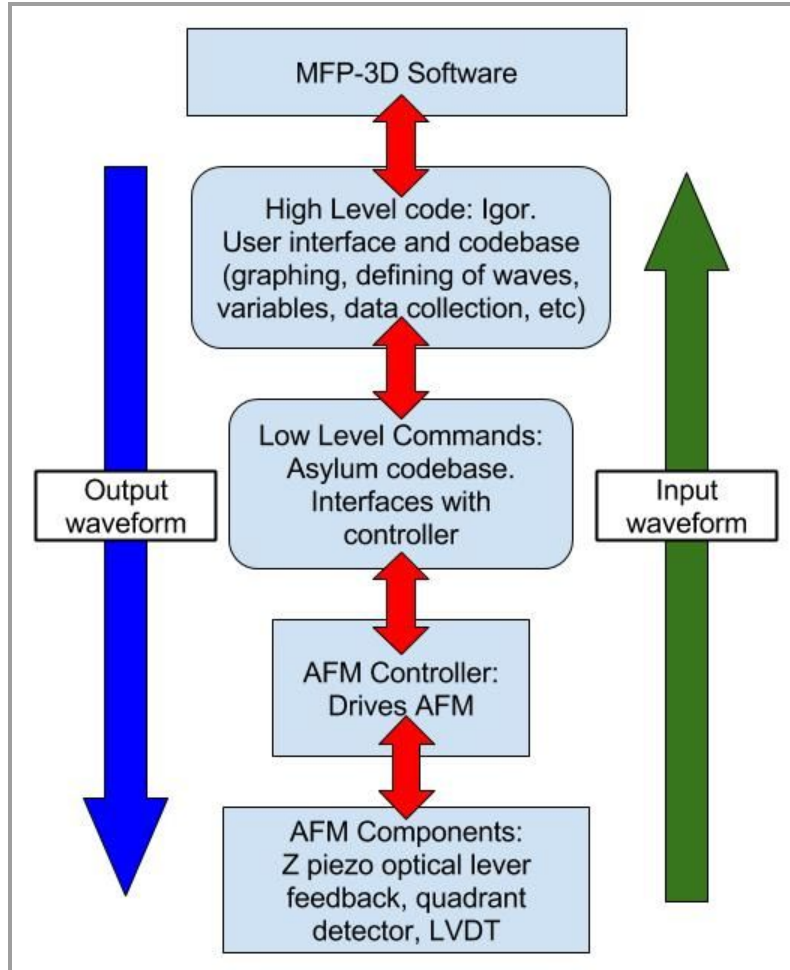


Figure 7: Diagram showing flow of output and input waveforms through each layer of the system related to the module.

2.2.1: Parameter Requirements

As previously stated, various output waveforms that are to be implemented as the impulse to the AFM cantilever. First, the chirp waveform. The impulses desired to be generated as impulses are white noise, square waves, chirp (frequency sweep), and kick (amplitude sweep). There are some requirements such as sampling frequency and maximum amount of data per unit time which are inherent to the Asylum system that need to be accounted for in parameter design for the waveforms. The sampling frequency should be in accordance with the Nyquist frequency

of the system. The ADC and DAC of the signal processing components of the AFM operate at 100 kHz. The frequency of the output will consider this to avoid aliasing.

The output waveforms will use the low level function “td_XSetOutwave”. Similarly, the response of the system will be input to the computer using the low level function “td_XSetInwave”. These functions will take arguments and data which are defined in the high level IGOR code, and return data which is to be manipulated in the IGOR software.

The safety of the machine must be considered when exciting the tip, especially when it is spatially extremely close to the sample. Tips will be broken very easily if it is not understood how much room is available to work with for a given impulse- if the displacement of the tip is relatively output to go past the surface material by too large an increment, not only will the tip break but also other components. AFM tips are costly and other components in the machine are cripplingly expensive to replace.

With respect to distance of tip to surface, the tip should be able to be put into attractive and repulsive regimes while the user defined software runs. Thus, the custom software should have the capability of being called when the machine is in normal operation.

2.2.2: Reproducibility and User Guide

The entire process of code development should be documented in a user guide. Future users will have this user programming guide as a resource if this project is to be continued or reproduced in the future. A proper guide will aid future development by limiting background research time into the code structure of the Asylum system. Furthermore, an Asylum user forum exists for user programming which contains a fair amount of open source programming resources.

2.3: Functional Decomposition

2.3.1: Code Organization

With the design specifications listed above, the software should be designed in a modular way. A block diagram is shown below of the interaction between different pieces of the module. This design will allow for the removal or addition of different waveform subfunctions. Each of

the subfunctions are delegated the task of performing the low level AFM interactions, while the Init main program defines parameters and the impulse waveform. The data collected is allocated to a separate high level IGOR program which will perform data analysis and organize the impulse response data. Figure 8 details a block diagram of the organization of each program in the overall module.

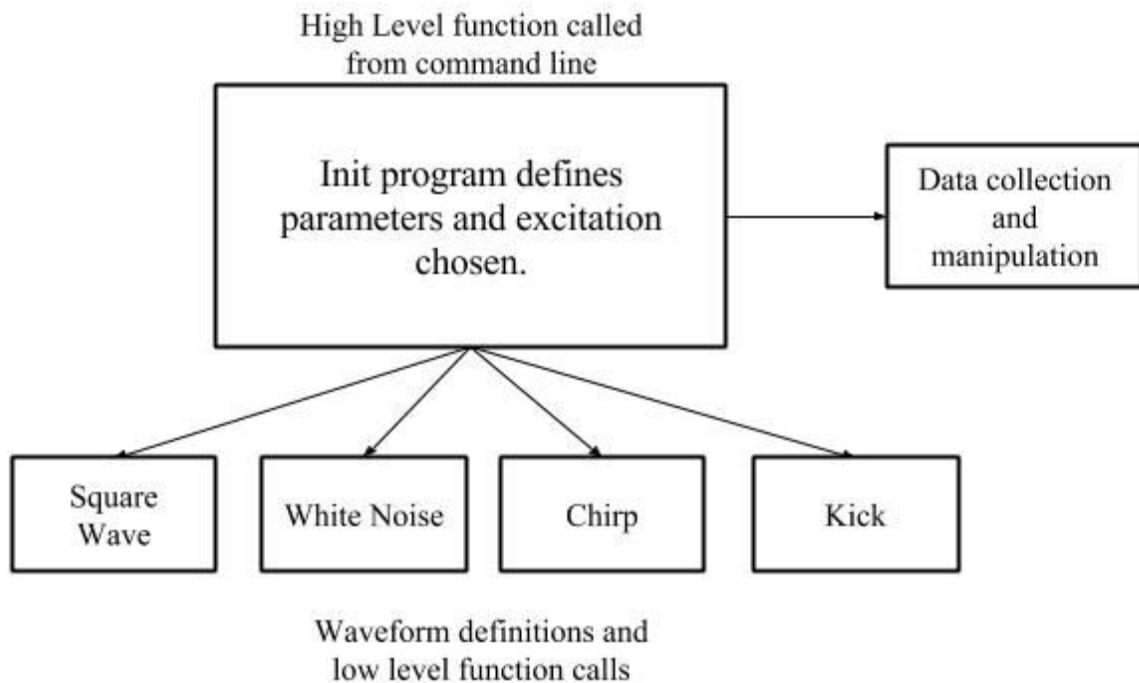


Figure 8: Block Diagram of code organization

2.3.2: Selection of Design Criterion

As this software module has been created concurrently with a detailed document and guide to reproduce the results and add onto the module, much of the design criterion selection is outlined step by step. Gaining access to the ability to produce and compile user defined functions to be in use with an operating system was a difficult task, but will be able to be reproduced easily

in the future by other projects by means of thorough documentation. Hierarchical organization of the code also helps for reproducibility.

For this impulse response study, data should be collected in an organized manner. With only small variations to the main init file, the parameters of different excitations should be easily modified. Data collection will be organized into folders by data and saved as waveform graphs and tabular data. As the outputs are digital, they can be reproduced in various regimes with various sample types to allow for a systematic study of the impulse response with a thorough amount of data.

Section 3: Design Alternatives

While the objective of this project in regard to requirements of the impulse response software module is straightforward, there were some pertinent choices that were considered in the structure of the module. There are different ways of measuring the displacement of the AFM tip which offer different information about the system. Furthermore, the means of exciting the AFM tip had alternative options and a large variety of impulse waveforms to choose from of differing complexity to model computationally.

3.1: Z-dimension Sensor Readings

The displacement of the AFM tip in response to the input excitation can be measured in two ways. For all dimensions of AFM tip movement, the displacement and relative position is measured by the Asylum LVDT sensor. For fine motion in the z-dimension, there is the photodetector - a photosensitive diode (PSD) In the case of this project's interest, the sensitivity of the z-dimension movement is of most interest, and the advantages of using either of these options is considered. A diagram from the AFM Manual is included below showing the layout of the AFM head with the Z LVDT and PSD relative to other components of the AFM head. [2]

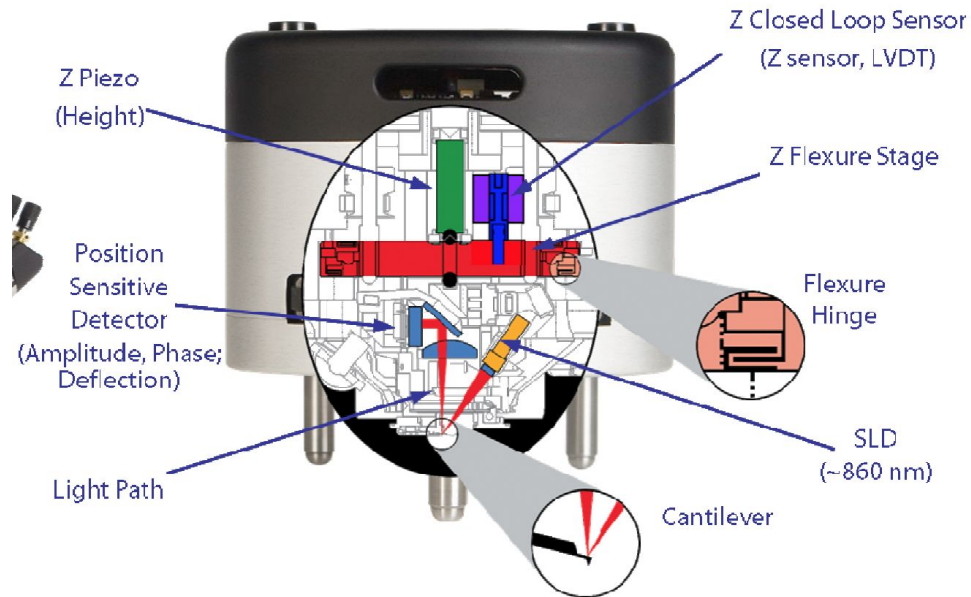


Figure 9: Diagram showing the different components involved with the excitation of the AFM tip.

3.1.1: LVDT Voltage Output

The z dimension LVDT sensor operates on a closed loop, meaning that it has a set voltage (displacement, force) at which it maintains the tip distance from the sample. Operating on a feedback loop, the change in voltage returned from the digitized value can be read in by Asylum software. This displacement measured is a quantity related to the cantilever and tip movement, but by itself does not contain enough information to detail the fine movement of the impulse response, and thus is not the best option for our design criteria.

The LVDT has a factory specification of measuring within a 0.7 nm precision. It is a measure of the flexure stage, not of the direct response of the tip itself. Thus, it cannot act as the primary source of data for the impulse response. It can be used to quantify the system response of the impulse to the stage, which can be useful for other purposes.

The LVDT sensor in Asylum software contains two parameters which can be read in from the user function. It contains a ‘raw’ signal, but according to data specifications this

channel is too noisy to gather meaningful data. The LVDT ‘clean’ signal which will be used to gather information from the software module is processed with the following procedure to produce viable data:

1. Each section of the voltage displacement plot is broken down and separated. This is made up of the extension and the retraction of the z piezo.
2. Fits a 7th order polynomial to the extend plot.
3. Fits a 5th order polynomial to the retract plot.

The data from the LVDT will be used in its ‘clean’ signal. It can be used complementary to the data from the optical lever and PSD, but it is likely that it will not be an ideal representation of the measure of response of the tip to an impulse. [2]

3.1.2: Optical Laser Photodetector: PSD Output

Optical laser photodetector, in the MFP-3D AFM implemented as a photosensitive diode (PSD), is the highest level of acute clarity which can be attained by the microscope about the cantilever and tip. Also operating on a closed loop, the amount of deflection on a four-quadrant photodetector can be measured. The amount of deflection can be used in feedback loop to alter the output signal to the tip. Figure 10 is a diagram provided by the AFM manual is included below, illustrating the operation of the PSD detection and complementary components. This diagram also shows important detail about AFM design which should not be omitted in this report.

The SLD (superluminescent diode) produces a laser which reflects off the back of the cantilever, which is reflected onto the PSD. The PSD measures the displacement of the cantilever in units of nm/V and offers fine detail about the movement of the cantilever and tip which the LVDT cannot offer. This signal, which also has the benefit of having a 5MHz ADC compared to the LVDT’s 100 kHz ADC, is clearly a more robust measure for the impulse response of the system. In data collection, the PSD measure of cantilever and tip displacement will be the primary source of data which will show the impulse response of the system. [2]

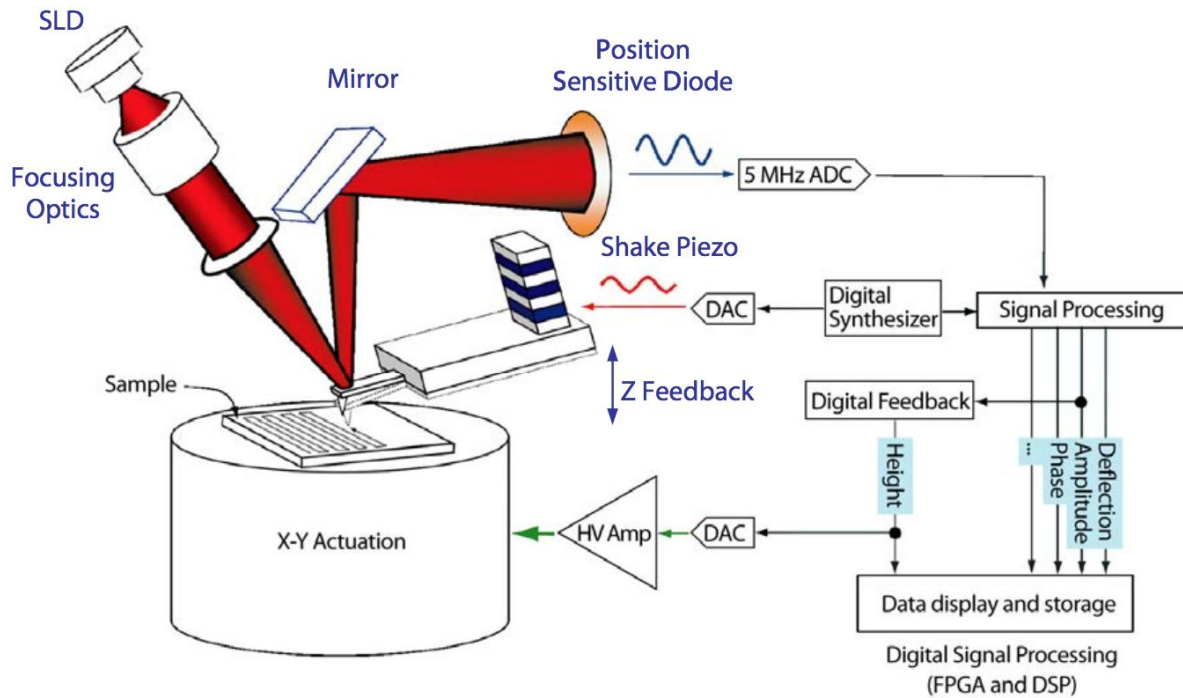


Figure 10: Overall AFM operation [2]. Detail is shown in describing the operation of the PSD interaction with the rest of the system.

3.2: Excitation Strategies

3.2.1: Sonification

Considering the past work of Matt Manning '12, it might seem reasonable to try and reproduce his findings with the sonification impulse response of the AFM tip. However, that experiment served primarily as more of a validation that a systematic study using AFM software to study the impulse response has potential to yield interesting results. The method of using an audio signal to indirectly excite the AFM tip doesn't offer as much clarity in control of experiment reproducibility. [5]

3.2.2: Choice of Excitation Output

Different waveforms are desired to choose from to use as the output excitation impulse. Some waveforms to choose from for the cantilever and tip excitation are detailed in the following table. These output signals vary in their built in ability to be produced in IGOR and

will likely all be considered for use in the study of system response. Note that all of these excitations rely on the same max frequency of 50kHz.

Excitation Waveform	Difficulty to produce in IGOR	Mode of wave production
Square Wave	Moderate	User defined procedure
White noise	Easy	Built in IGOR function
Chirp	Difficult	Import wave data
Kick	Difficult	Import wave data

Table 1: Excitation waveform options.

Other excitations were considered such as DC step increments, however this system response should be produced with the square wave. The chirp and kick waveform are more complicated signals that can be produced in MATLAB. The wave data can be imported into IGOR, but they are difficult to produce relative to the other excitations.

The final design alternative considered was the means of outputting and reading data to and from the AFM in software. The built in low level Asylum code only provided one realistic function to perform this task, so there was no design choice other than ‘td_XSetOutwave’ and ‘td_XSetInWave’ functions. Without these built in functions, the project may not be possible. The only other way to excite the AFM tip is by direct voltage changes, which would not produce an impulse response that is measureable.

Section 4: Preliminary Proposed Design

With the background and design requirements thoroughly considered and determined, a

final proposed design for the term is presented next. The reader should note that the previous sections set the stage for this final one, detailing every element of the design which will be implemented. Thus, this section will attempt to be terse and organized. This section will state the strategy of meeting all of the specified design requirements and the current status of the module design.

4.1 Software Design High Level Breakdown:

The following list is the high level action item approach to software development of the module. Much of the following has been in process this term.

1. Gain ability to implement user defined functions. Create user defined output of a waveform to excite cantilever with impulse. Document process and create user manual. Confirm ability to compile and run programs which access AFM component parameters while in normal operation. Find high level and low level functions which will be helpful in implementing module, find corresponding documentation. Identify AFM components which will be altered by software and find component part information in order to find which components can be measured to record impulse response.
2. Define output and input parameters within impulse software module. Parameters include frequency range, sampling frequency, and distance from tip to sample. Implement consistent parameter definition approach for each waveform subfunction, begin to write routines to generate each impulse output waveform. Maintain code structure and documentation.
3. Test validity of software excitation. Find input frequency compared to measured frequency and other such possible inconsistencies. Look for improvements on scientific details of interest like frequency range with regard to tip resonant frequency. Software module will be able to allow for systematic study of impulse response of cantilever tip for various output waveforms.

4.2 Testing/ Characterization:

The following list details the areas of implementation which are to be explored next term. Item 1 picks up where the final item in the last list ends.

1. Complete functions, test validity. Establish strategy for writing and altering programs in modular manner.
2. Start testing in free space. Once confident in impulse functions, move to attractive and repulsive regimes to test impulse response. [6]
3. Systematic study of impulse response using software module. With each excitation, test in attractive and repulsive regimes with increasing amplitude. Collect data with different samples.
4. Study impulse response. There is reason to believe that there will be an observed nonlinear system based on prior work.

4.3 Project Timeline:

Fall 2017: Implement prototype programs that allow for user defined excitation of the AFM cantilever for impulse, chirp, white noise, and square wave. Ability to control height of tip above sample when performing excitations. Record input waveform of quadrant detector. Record optical lever output and understand relationship to cantilever motion. Consider all calibrations that will factor into the output of excitation and recording of data: F_s , distance of tip to the sample, define system thresholds to avoid damaging machine.

Winter 2018: Using user defined functions, complete systematic measurements for different excitations with increasing amplitude. Explore expected nonlinearity of system. Comparison of these measurements to corresponding audio experiment. Comparison in attractive and repulsive regimes. Characterize the activation and response corresponding to each excitation and parameter setting. Time characteristics and Fourier Transform measured.

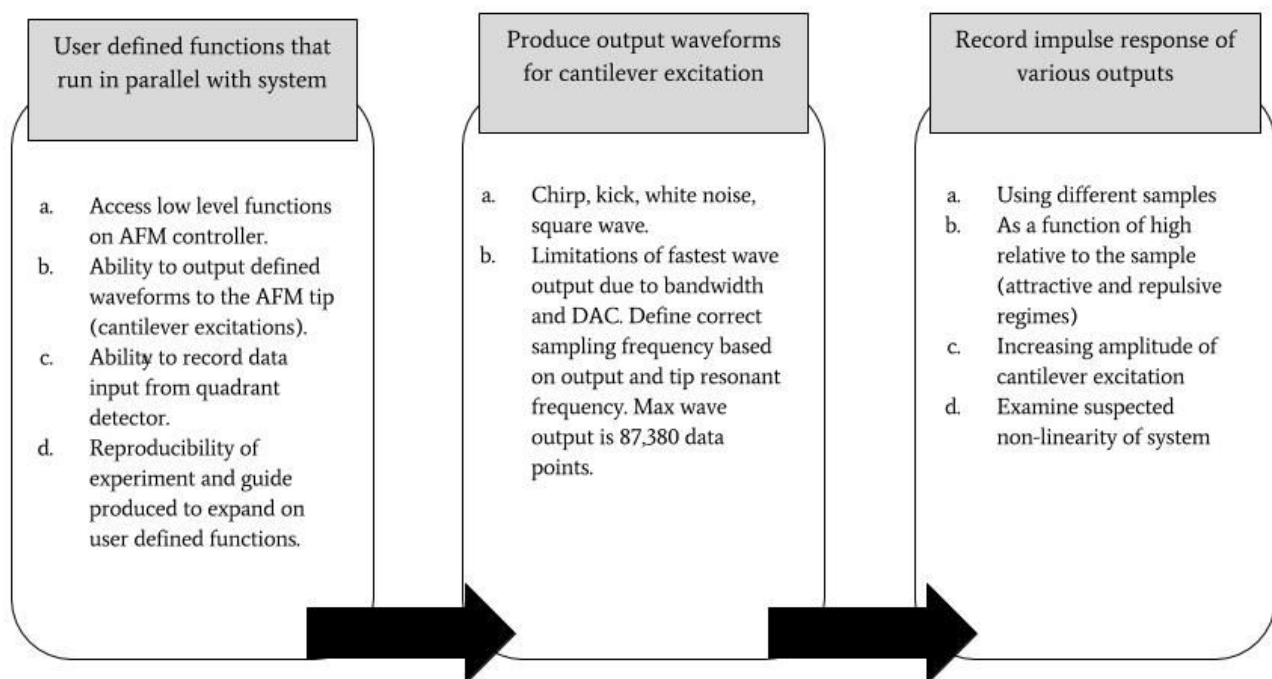


Figure 11: Design workflow diagram

4.4: Parts List

Vendor	Product Name	Qty	Price	S&H
Asylum Research	Econo-LTESP-Au (AFM Tips)	10	\$210	\$5
Asylum Research	HQ-300 AU (AFM Tips)	10	\$200	\$5

Table 2: Parts List. Note that different tips were made available courtesy of the Prof Samuel Ammanuel of the Physics Dept. This parts list reflects the request for funding from the SRG

4.5: Status of Module Design

The following table states the status of major milestones in the development of the impulse response module.

Design Requirement	Current Status	Details	Impact of Progress
Find low level functions to interface with cantilever and tip	COMPLETE	“Td_” commands allow for direct interface with x, y, and z piezo fine control. Output and input waveforms can be input to components. Input waveforms of response can be read in similar manner.	The low level controller commands are well documented. The specific routines “td_xsetoutwave” and “td_xsetinwave” are the primary functions to output and record the impulse and respective response. The main task of the module is made possible by these low level commands.
Interface high level IGOR code to create waveform data and low level functions	COMPLETE	The Asylum user forums and programming guide show examples of how to interface with controller low level commands.	The high level IGOR interface is very malleable for future changes and is robust in documentation and capability. The high and low level programming capabilities of the system have proven to allow for the major desired design requirements to be

			completed in the software suite.
Determine Maximum Sampling frequency for output/ input waveforms	COMPLETE	The A/D and D/A converters run at 100 kHz. Maximum sampling frequency is 50 kHz. For the PSD, the A/D conversion runs at 5MHz.	The maximum sampling frequency of 50 kHz is disappointing and is a limiting factor moving forward. The full frequency range of AFM tips are desired to be considered in the excitation waveforms. Many AFM tips have a resonant frequency greater than 50 kHz.
Gathering data for cantilever and tip response: LVDT and Quadrant detector	TODO	The LVDT sensor for the z-dimension voltage reading is readable as a waveform. The quadrant detector data is not as accessible, and as of now is only readable as a single displacement value.	The quadrant detector data is essential to analyzing the response of the cantilever and tip to the impulse response. The LVDT sensor does not reveal as much detail about the tip's movement. Reading quadrant detector data as an input waveform is an ongoing challenge.
Impulse response to be studied as a function of height	TODO	The distance of the tip relative to the sample is not	Attractive and repulsive regimes correspond to a phase reading that is only

relative to sample		addressed in the software suite. It is likely that this will not be a feature of the code, but rather input at runtime.	available to fine tune during run time. As implementation has not yet begun, there has not been an opportunity to address this requirement.
Subfunction creation for each output excitation	COMPLETE	All desired excitation waveforms: the kick, chirp, noise, and square waves all have completed subfunctions.	IGOR can input MATLAB data, and the desired waveforms can be easily created in MATLAB.
Data processing: Impulse response data analysis	TODO	For analyzing the impulse response data, all that is complete in the software suite is the creation of a graph of z dimension movement. This data must be made tabular and discrete for further analysis.	The analysis of data will not be an issue within the confines of IGOR. This design requirement has not yet been addressed as implementation has not yet begun.

Table 3: Progress on module (as of Fall term 2017)


Section 5: Final Design and Implementation

This section will present all working parts of the final design. The completed programs of the impulse response module will be explained. Pseudocode will be included for each completed program of the software module. Descriptions of the exact high level (IGOR) and low level (Asylum controller) functions utilized will be included. Note that within the user manual for this project there exists a more detailed outline of how to use and manipulate the completed software.

5.1 Input and output waveforms

Matching the design criteria, the most important part of the software module's functionality is to have user defined excitation waveforms drive the z-piezo while also recording the deflection and z-piezo sensor values as input waveforms. There are a specific sequence of high level IGOR commands and low level Asylum controller commands which successfully implement this design requirement. See pseudo code below for a high level look at how the software must operate. In the following sections, each enumerated point will be detailed.

Implementation method: pseudocode

1. Stop all current controller activity which may harm the system.
 2. Define variables such as number of data points to be output and amplitude of excitation.
 3. Create waveform data variables
 4. Read current Z piezo voltage. This is the current height of the Z piezo.
 5. Define the impulse drive wave for use in cantilever excitation.
 6. Send Output wave to controller
 7. Define the input impulse wave to be recorded
 8. Trigger the output and input waves to the same event.
 9. Display wavedata (output excitation and impulse response)
- 

5.1.1: Stopping controller activity

Very important to consider when writing custom programs for the AFM is the safety of the system. The first task is thus to ensure that no functions are currently driving the machine which may damage the AFM head.

This task is quite simple and is completed with two low level Asylum controller “td” commands. The first is “td_stopThermal()”, which stops all thermal data collection on the head. Directly following this, the command “td_stop()” is used. All inputs, outputs, and feedback loops on the controller are now halted, and the system is in an idle, static state. This is the required state for our excitation function to operate at.

It is important to clear the controller before executing custom functions on it. Based on the previous state of the AFM, it is possible to do damage to the head. It is also important to use the “td_stopThermal” command before using the “td_stop” command, because using “td_stop” to clear thermal data collection can leave the controller in an error state.

5.1.2: Defining variables

Next, the high level IGOR variables need to be declared. The variables used are as follows:

1. ScanRate: sets the amount of time for the excitation. $1/\text{ScanRate} = \text{time of excitation}$.
2. startZVoltage: this will store the z-piezo voltage which the sensor reads at the time before the excitation.
3. A: amplitude of the excitation waveform.
4. dataPoints: The number of data points for the output and input waveforms. The number of points in the wave must be a multiple of 32. All waves on the same “bank” must be of the same length. The controller memory bank and waveform channels are explained in section 5.1.5.

5.1.3: Create waveform data variables

Using high level IGOR code, memory must be allocated in the creation of the output and input waveforms. By convention for Asylum low level code, these waves must have an integer value of data points which is a multiple of 32. Syntax for their creation is straightforward in IGOR, and their initialization requires only that they are declared and displayed to the screen.

5.1.4: Read current z-piezo voltage

This is another simple task in the sequence of the program. Using the ‘startZVoltage’, assign the current value of the z-piezo to this variable. This is completed using the low level command “td_ReadValue” and using the “Head.Input.Z” as the parameter. This is an important value as it will determine the range of the excitation amplitude which will drive the z-piezo. An alternative to reading the z-piezo voltage would be to set the voltage, but the approach employed in this program logic is to take the input sensor reading.

5.1.5: Defining output waveform for controller

The Asylum command which allows for output waveforms to excite AFM components is the “td_xSetOutWave”. Similarly, the “td_xSetInWave” command records the component sensor values. These are the most important commands involved with the software for the purpose of this project. The parameters for each function, and their corresponding significance for controlling AFM hardware essential to understanding system operation. The basic syntax for the “td_xSetOutWave” low level command is below, with explanation of the parameters following this.

```
td_xSetOutWave(whichBank, eventString, channelString, wave, interpolation)
```

whichBank determines which memory bank in the controller’s DSP the

wave will be stored in. There are three banks of memory which can each hold up to two waveforms. The waves can be up to 87,380 data points and must be single-precision, 32 bit floating point. Because the purpose of this module is to have a single output excitation drive the z-piezo, this value will be “0”. The excitation waveform is stored on channel 0, with the two input waveforms for z-piezo sensor values and cantilever deflection values stored on channel 1.

eventString determines when the data will be output. This is the trigger for when the wave will be output to a given AFM component. In the case of the impulse response module, all input and output waveforms are to be triggered at the same time. The input for this parameter will thus be “ARC.Event.0”.

channelString is the low level parameter name that names the desired AFM component. For the output excitation waveform the string will thus be “ARC.Output.Z”, so the z-piezo will be specified to be driven by the waveform.

wave is the name of the IGOR wave that will be sent to the controller. For the output waveform, it will be the specified excitation waveform.

Interpolation determined how quickly the wave will be output. The DAC’s and ADC’s for the z-piezo run at 100 kHz, so an interpolation of 1 means that each point in the IGOR wave will correspond to a 100 kHz sample. We use a 1024 point wave for excitations as the baseline, so with an interpolation of 1, the wave will be output in about 10 ms.

Combining this syntax with the input waveform syntax will give the following:

Output z piezo excitation	td_xSetOutWave(0, “ARC.Event.0”,”ARC.Output.Z”, DriveWave,
---------------------------	--

	$\text{round}(\text{cMasterSampleRate} / (\text{ScanRate} * \text{DimSize}(\text{DriveWave}, 0)))$
Input z sensor reading	<code>td_xSetInWave(1, "Event.0", "ARC.Input.Z", DataWaveZ, 1)</code>
Input cantilever deflection reading	<code>td_xSetInWave(1, "Event.0", "Deflection", DataWaveDefl, 1)</code>

The major specifications which are involved with the software as listed below:

1. Controller ‘channels’: Due to bandwidth limitations, each of the three channel banks have different specifications:
 - a. Bank 0: two 32-bit waves at 100 kHz
 - b. Bank 1: two 32-bit waves at 100 kHz
 - c. Bank 2: one 32-bit wave at 50 kHz
2. IGOR excitation waveform: this is defined in software as some waveform, determined by the user. For example, we can use a simple sin function. This is declared in the high level software.
3. Z-piezo voltage range: can go from -10 to +150 V. Any value which is output as an excitation outside of this range will result with clipping in the resulting z sensor reading.

Performance estimates and results

Present the estimated performance of the project (and how they were derived) based on the preliminary design (estimates to include speed, cost, power consumption, noise-immunity, ease of use, etc, depending on the project). Present the actual performance results. Discuss the results,

compare with estimated performance and explain discrepancies. Include suggestions for design changes that would improve the performance of the project. Use graphs or other figures to show relationships when appropriate.

Discussion of data collected:

Looking at the figure below, there are some major takeaways about the functionality of the software as well as potential for future data collection. The matrix of data below is organized by means of using rows for each different system response. For example, the waveform in the top left, in the “A” row is under the “Output waveform” column. This is the excitation which is defined in software. Next to it on the right, still in the “A” row, under the “Z piezo sensor input waveform” is the input sensor reading from the z-piezo which corresponds to the excitation. In the top right, still in row “A”, under the “Deflection input waveform”, is the deflection value which corresponds to the excitation. In this way, the data is organized. Included are the results of 5 different types of excitations to the cantilever in this manner. All values are in the time domain. For all data, a AC240 cantilever with resonant frequency of 70 kHz was used.

The two programs tested are the sinusoidal excitation (rows of data A, B, and E) and the step response excitation (rows of data C and D). The programs are explained in the previous section, so the only specifications required for detailing these experiments are the values used for the excitation parameters.

1. Row “A”

C/6X'èqup'g'gze kwqp

- i. Note that the z piezo sensor is reading nearly the exact same values as the output waveform.
- ii. The deflection of the cantilever and tip matches the waveform, but is an inverted signal. There is a small amount of clipping, which is explored in B.

2. Row “B”

D/! X'èqulpg'gz ekcvkqp

- i. More pronounced clipping is observed in both the z-piezo sensor and cantilever response
- ii. The z-piezo has a voltage input limit ranging from -10V to 150V. When the input waveform goes below -10V, we the tip should no longer move. Clipping is expected.

3. Row “C”

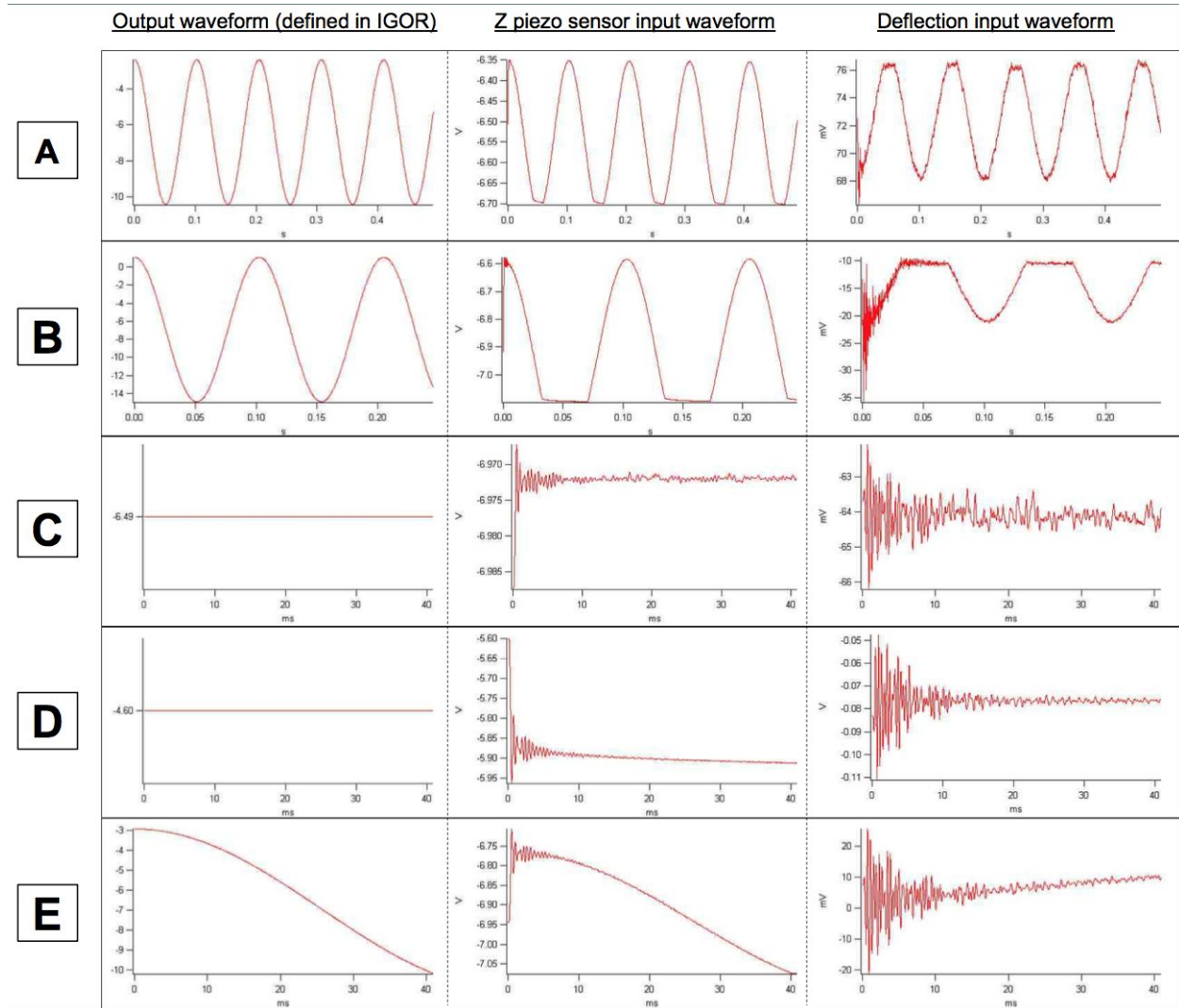
- a. This is a 0.5V step response. It is important to note that the starting voltage is a residual value from whatever value the z-piezo was currently set to when the program started running. Note that there the response in the z sensor and cantilever deflection shows transient effects.

4. Row “D”

- a. This is a 1V step response. Again, there is a transient response in the z sensor and deflection readings. In this case, the excitation was performed with the starting z voltage at the previous resulting value from the response in row C.

5. Row “E”

- a. This is a sinusoidal response of a 4V cosine waveform. We again see a transient response.



The most important aspect to take away from the results is that it is possible to record the response of system components to different user-defined excitations with the completed software. There is still work to be done with honing initial conditions of the system when executing the response software. It is likely that a reproducible way to solve this issue would be to set the z voltage to a constant starting value rather than running the software based on the past system state.

Production schedule

The general schedule of work is presented as a weekly outline. Following the production schedule chart, recommendations for improvements in scheduling and planning are included.

The Fall and Winter (498 and 499) are reviewed.

Schedule of work on Senior Capstone

Time Period	Goals	Tasks Completed	Comments
Spring Term 2017 (ECE 497)	Choose project	Custom software project for AFM chosen. Background work started Advisor: Prof Catravas	Co-advisor Katherine Dovidenko was added on to the end of the Spring term after starting off with just Prof Catravas
Fall Term 2017 (Weeks 1-3)	Explore Asylum research software	Documentation within the MFP-3D software scoured and studied. Online user forums were read to find similar user-defined functions and techniques	Getting acquainted with Asylum software took the majority of the Fall Term and would be part of project work until the beginning of 499
Fall Term 2017 (Weeks 4-7)	Begin writing simple code and taking benchmarks. Training on AFM.	Simple programs such as basic drive stage movement and measurement were written. Common compilation errors and IGOR syntax patterns were learned. AFM training sporadic.	It was a confusing process to get to the point where a single custom program could be successfully executed. The user manual should contain the wisdom gained from this difficulty.
Fall Term 2017 (Weeks 8-10)	Work on developing impulse response software module	Input and output waveform program was written, compilation	The only results that could be read out from the system was noise from the

		successfull but contained many errors.	components. Software was not driving cantilever with excitation yet.
Winter term 2018 (Weeks 1-5)	Work on developing impulse response software module	Benchmarks such as confirming output excitation was not in sync with input response waveform were recorded.	Further exploration of system parameters followed observed errors in system response to software.
Winter Term 2018 (Weeks 5-10)	Focus work on sinusoidal excitation and step response programs	Both programs completed.	Persisting problems were addressed and observable and accurate data collected.

Improvements that could be made in the development of this senior project include more rigorous documentation of experiments completed with the accompanying software. Often, unsuccessful software was edited and deleted periodically, so the accompanying results of component data has no corresponding code. This led to attempts at fixing software bugs to be single minded shots in the dark rather than incremental.

User's Manual*

This project features an extensive user manual which was completed as part of the Union College Scholar's program work study. It is broken into two parts which will benefit future users of the Union MFP-3D:

1. A guide for general Asylum codebase manipulation.
2. How to employ completed code and control system parameters for study of AFM tip impulse response. *This is a user manual for the software completed in this project.

This user manual is included as a supplement to this report and can be used as a separate resource for similar projects in the future.

Discussion, Conclusions, and Recommendations

A software module has been written that allows for sinusoidal and step excitation of the Asylum AFM z-piezo with readout of cantilever deflection and z-sensor response. The output tracks the input frequency correctly and shows transient effects. Gradual progress was made throughout the course of ECE 498 and 499 which allowed for the benchmarked development of user defined functions on Union College's Asylum Research MFP-3D AFM. Testing of the software module is currently in progress, with future work to include further calibration of impulse response input parameters. With the software module mostly completed, it will be the work of a future project to define the type of excitation waveform to be sent to the AFM's tip in order to study the impulse response systematically for a variety of samples.

Roadblocks which impeded the development of this project included a limited amount of documentation for Asylum low-level functions as well as a small pool of available resources for IGOR programming techniques. In order to remedy these challenges for any future work, my programming guide for the AFM should act as an organized congregation of custom AFM programming resources. The wide scope of this project involved a substantial amount of background learning and training to understand the AFM before gaining the ability to work on it. For this reason, it would be recommended that future work start with AFM training in conjunction with detailed learning about relevant system components which the impulse response software module can manipulate. There exist a wide variety of studies which can be explored with the software module which has been developed.

Acknowledgements

I would like to thank the following people for helping make the design of this project possible:

- Advisors Palmyra Catravas and Katherine Dovidenko
- Union College ECE Department
- Samuel Ammanuel, Union College Physics Dept
- Maxwell Stwertka, Union '18
- Asylum Research
- WaveMetrics (IGOR documentation)

References

- [1] Eaton, Peter Jonathan, and Paul West. Atomic Force Microscopy. Oxford University Press, 2014.
- [2] Asylum Research. Ryan Fuierer, Ph.D. (2009). Procedural Operation Manual. Version 10.5
- [3] Asylum Research. (2010). Basic Programming Tutorial User Guide. Rev. 552, Dated 5/21/2010
- [4] F.L. Leite, L.H.C. Mattoso, The Atomic Force Spectroscopy as a Tool to Investigate Surface Forces: Basic Principles and Applications, Modern Research and Educational Topics in Microscopy (2007)
- [5] Manning, Matthew, Coupling of audio signals into AFM Imaging, Union College Senior Project, 2012
- [6]U. Rabe, J. Turner, Analysis of the high-frequency response of atomic force microscope cantilevers, Appl. Phys. A 66, S277–S282 (1997)

